Scoring Lexical Entailment with a Supervised Directional Similarity Network

Marek Rei[♠♢] **Daniela Gerz**[♣] **Ivan Vulić**[♣]
[♠]Computer Laboratory, University of Cambridge, United Kingdom
[♢]The ALTA Institute, University of Cambridge, United Kingdom
[♣]Language Technology Lab, University of Cambridge, United Kingdom
marek.rei@cl.cam.ac.uk, dsg40@cam.ac.uk, iv250@cam.ac.uk

Abstract

We present the Supervised Directional Similarity Network (SDSN), a novel neural architecture for learning task-specific transformation functions on top of general-purpose word embeddings. Relying on only a limited amount of supervision from task-specific scores on a subset of the vocabulary, our architecture is able to generalise and transform a general-purpose distributional vector space to model the relation of lexical entailment. Experiments show excellent performance on scoring graded lexical entailment, raising the state-of-the-art on the HyperLex dataset by approximately 25%.

1 Introduction

Standard word embedding models (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) are based on the distributional hypothesis by Harris (1954). However, purely distributional models coalesce various lexico-semantic relations (e.g., synonymy, antonymy, hypernymy) into a joint distributed representation. To address this, previous work has focused on introducing supervision into *individual* word embeddings, allowing them to better capture the desired lexical properties. For example, Faruqui et al. (2015) and Wieting et al. (2015) proposed methods for using annotated lexical relations to condition the vector space and bring synonymous words closer together. Mrkšić et al. (2016) and Mrkšić et al. (2017) improved the optimisation function and introduced an additional constraint for pushing antonym pairs further apart. While these methods integrate hand-crafted features from external lexical resources with distributional information, they improve only the embeddings of words that have annotated lexical relations in the training resource.

In this work, we propose a novel approach to leveraging external knowledge with general-purpose unsupervised embeddings, focusing on the directional graded lexical entailment task (Vulić et al., 2017), whereas previous work has mostly investigated simpler non-directional semantic similarity tasks. Instead of optimising individual word embeddings, our model uses general-purpose embeddings and optimises a separate neural component to adapt these to the specific task. In particular, our neural Supervised Directional Similarity Network (SDSN) dynamically produces task-specific embeddings optimised for scoring the asymmetric lexical entailment relation between any two words, regardless of their presence in the training resource. Our results with task-specific embeddings indicate large improvements on the HyperLex dataset, a standard graded lexical entailment benchmark. The model also yields improvements on a simpler non-graded entailment detection task.

2 The Task of Grading Lexical Entailment

In graded lexical entailment, the goal is to make fine-grained assertions regarding the directional hierarchical semantic relationships between concepts (Vulić et al., 2017). The task is grounded in theories of concept (proto)typicality and category vagueness from cognitive science (Rosch, 1975; Kamp and Partee, 1995), and aims at answering the following question: *"To what degree is X a type of Y?"*. It quantifies the degree of lexical entailment instead of providing only a binary *yes/no* decision on the relationship between the concepts X and Y, as done in hypernymy detection tasks (Kotlerman et al., 2010; Weeds et al., 2014; Santus et al., 2014; Kiela et al., 2015; Shwartz et al., 2017).

Graded lexical entailment provides finer-grained

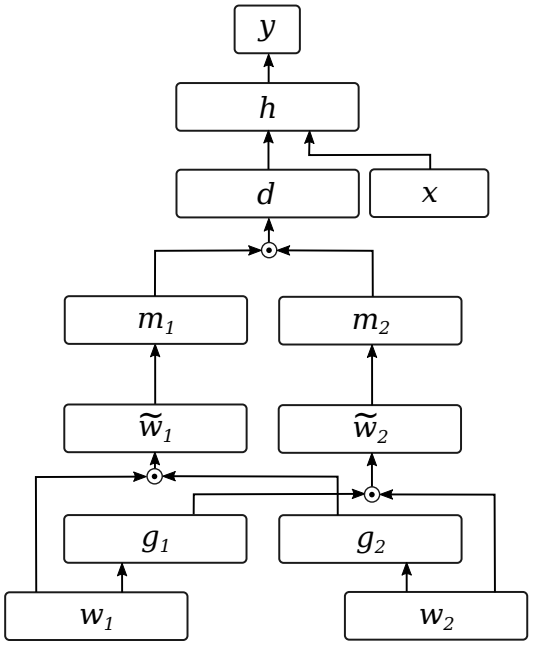

Figure 1: Supervised directional similarity network (SDSN) for grading lexical relations.

judgements on a continuous scale. For instance, the word pair (*girl* → *person*) has been rated highly with 9.85/10 by the HyperLex annotators. The pair (*guest* → *person*) has received a slightly lower score of 7.22, as a prototypical guest is often a person but there can be exceptions. In contrast, the score for the reversed pair (*person* → *guest*) is only judged at 2.88.

As demonstrated by Vulić et al. (2017) and Nickel and Kiela (2017), standard general-purpose representation models trained in an unsupervised way purely on distributional information are unfit for this task and unable to surpass the performance of simple frequency baselines (see also Table 1). Therefore, in what follows, we describe a novel supervised framework for constructing task-specific word embeddings, optimised for the graded entailment task at hand.

3 System Architecture

The network architecture can be seen in Figure 1. The system receives a pair of words as input and predicts a score that represents the strength of the given lexical relation. In the graded entailment task, we would like the model to return a high score for *(biology → science)*, as biology is a type of science, but a low score for *(point → pencil)*.

We start by mapping both input words to corresponding word embeddings w_1 and w_2. The embeddings come from a standard distributional vector space, pre-trained on a large unannotated corpus, and are not fine-tuned during training. An element-wise gating operation is then applied to each word, conditioned on the other word:

$$g_1 = \sigma(W_{g_1} w_1 + b_{g_1}) \quad (1)$$
$$g_2 = \sigma(W_{g_2} w_2 + b_{g_2}) \quad (2)$$
$$\widetilde{w}_1 = w_1 \odot g_2 \quad (3)$$
$$\widetilde{w}_2 = w_2 \odot g_1 \quad (4)$$

where W_{g_1} and W_{g_2} are weight matrices, b_{g_1} and b_{g_2} are bias vectors, $\sigma()$ is the logistic function and $\odot$ indicates element-wise multiplication. This operation allows the network to first observe the candidate hypernym w_2 and then decide which features are important when analysing the hyponym w_1. For example, when deciding whether *seal* is a type of *animal*, the model is able to first see the word *animal* and then apply a mask that blocks out features of the word *seal* that are not related to nature. During development we found it best to apply this gating in both directions, therefore we condition each word based on the other.

Each of the word representations is then passed through a non-linear layer with $tanh$ activation, mapping the words to a new space that is more suitable for the given task:

$$m_1 = tanh(W_{m_1} \widetilde{w}_1 + b_{m_1}) \quad (5)$$
$$m_2 = tanh(W_{m_2} \widetilde{w}_2 + b_{m_2}) \quad (6)$$

where W_{m_1}, W_{m_2}, b_{m_1} and b_{m_2} are trainable parameters. The input embeddings are trained to predict surrounding words on a large unannotated corpus using the skip-gram objective (Mikolov et al., 2013), making the resulting vector space reflect (a broad relation of) semantic relatedness but unsuitable for lexical entailment (Vulić et al., 2017). The mapping stage allows the network to learn a transformation function from the general skip-gram embeddings to a task-specific space for lexical entailment. In addition, the two weight matrices enable asymmetric reasoning, allowing the network to learn separate mappings for hyponyms and hypernyms.

We then use a supervised composition function for combining the two representations and returning a confidence score as output. Rei et al. (2017) described a generalised version of cosine similarity for metaphor detection, constructing a supervised operation and learning individual weights for each

feature. We apply a similar approach here and modify it to predict a relation score:

$$d = m_1 \odot m_2 \tag{7}$$
$$h = tanh(W_h d + b_h) \tag{8}$$
$$y = S \cdot \sigma(a(W_y h + b_y)) \tag{9}$$

where W_h, b_h, a, W_y and b_y are trainable parameters. The annotated labels of lexical relations are generally in a fixed range, therefore we base the output function on logistic regression, which also restricts the range of the predicted scores. b_y allows for the function to be shifted as necessary and a controls the slope of the sigmoid. S is the value of the maximum score in the dataset, scaling the resulting value to the correct range. The output y represents the confidence that the two input words are in a lexical entailment relation.

We optimise the model by minimising the mean squared distance between the predicted score y and the gold-standard score $\hat{y}$:

$$L = \sum_i (y_i - \hat{y_i})^2 \tag{10}$$

Sparse Distributional Features (SDF). Word embeddings are well-suited for capturing distributional similarity, but they have trouble encoding features such as word frequency, or the number of unique contexts the word has appeared in. This information becomes important when deciding whether one word entails another, as the system needs to determine when a concept is more general and subsumes the other.

We construct classical sparse distributional word vectors and use them to extract 5 unique features for every word pair, to complement the features extracted from neural embeddings:

- Regular cosine similarity between the sparse distributional vectors of both words.
- The sparse weighted cosine measure, described by Rei and Briscoe (2014), comparing the weighted ranks of different distributional contexts. The measure is directional and assigns more importance to the features of the broader term. We include this weighted cosine in both directions.
- The proportion of shared unique contexts, compared to the number of contexts for one word. This measure is able to capture whether one of the words appears in a subset of the contexts, compared to the other word. This feature is also directional and is therefore included in both directions.

We build the sparse distributional word vectors from two versions of the British National Corpus (Leech, 1992). The first counts contexts simply based on a window of size 3. The second uses a parsed version of the BNC (Andersen et al., 2008) and extracts contexts based on dependency relations. In both cases, the features are weighted using pointwise mutual information. Each of the five features is calculated separately for the two vector spaces, resulting in 10 corpus-based features. We integrate them into the network by conditioning the hidden layer h on this vector:

$$h = tanh(W_h d + W_x x + b_h) \tag{11}$$

where x is the feature vector of length 10 and W_x is the corresponding weight matrix.

Additional Supervision (AS). Methods such as retrofitting (Faruqui et al., 2015), ATTRACT-REPEL (Mrkšić et al., 2017) and Poincaré embeddings (Nickel and Kiela, 2017) make use of hand-annotated lexical relations for optimising word representations such that they capture the desired properties (so-called *embedding specialisation*). We also experiment with incorporating these resources, but instead of adjusting the individual word embeddings, we use them to optimise the shared network weights. This teaches the model to find useful regularities in general-purpose word embeddings, which can then be equally applied to all words in the embedding vocabulary.

For hyponym detection, we extract examples from WordNet (Miller, 1995) and the Paraphrase Database (PPDB 2.0) (Pavlick et al., 2015). We use WordNet synonyms and hyponyms as positive examples, along with antonyms and hypernyms as negative examples. In order to prevent the network from biasing towards specific words that have numerous annotated relations, we limit them to a maximum of 10 examples per word. From the PPDB we extract the Equivalence relations as positive examples and the Exclusion relations as negative word pairs.

The final dataset contains 102,586 positive pairs and 42,958 negative pairs. However, only binary labels are attached to all word pairs, whereas the task

requires predicting a graded score. Initial experiments with optimising the network to predict the minimal and maximal possible score for these cases did not lead to improved performance. Therefore, we instead make use of a hinge loss function that optimises the network to only push these examples to the correct side of the decision boundary:

$$L = \sum_i max((y - \hat{y})^2 - (\frac{S}{2} - R)^2, 0) \quad (12)$$

where S is the maximum score in the range and and R is a margin parameter. By minimising Equation 12, the model is only updated based on examples that are not yet on the correct side of the boundary, including a margin. This prevents us from penalising the model for predicting a score with slight variations, as the extracted examples are not annotated with sufficient granularity. When optimising the model, we first perform one pre-training pass over these additional word pairs before proceeding with the regular training process.

4 Evaluation

SDSN Training Setup. As input to the SDSN network we use 300-dimensional dependency-based word embeddings by Levy and Goldberg (2014). Layers m_1 and m_2 also have size 300 and layer h has size 100. For regularisation, we apply dropout to the embeddings with $p = 0.5$. The margin R is set to 1 for the supervised pre-training stage. The model is optimised using AdaDelta (Zeiler, 2012) with learning rate 1.0. In order to control for random noise, we run each experiment with 10 different random seeds and average the results. Our code and detailed configuration files will be made available online.[1]

Evaluation Data. We evaluate graded lexical entailment on the HyperLex dataset (Vulić et al., 2017) which contains 2,616 word pairs in total scored for the asymmetric graded lexical entailment relation. Following a standard practice, we report Spearman's ρ correlation of the model output to the given human-annotated scores. We conduct experiments on two standard data splits for supervised learning: *random split* and *lexical* split. In the random split the data is randomly divided into training, validation, and test subsets containing 1831, 130, and 655 word pairs, respectively. In the *lexical split*, proposed by Levy et al. (2015), there is no lexical overlap between training and test subsets. This prevents the effect of *lexical memorisation*, as supervised models tend to learn an independent property of a single concept in the pair instead of learning a relation between the two concepts. In this setup training, validation, and test sets contain 1133, 85, and 269 word pairs, respectively.[2]

[1] http://www.marekrei.com/projects/sdsn

	Random		Lexical	
	DEV	TEST	DEV	TEST
FR	-	0.299	-	0.199
SGNS-DEPS	-	0.250	-	0.253
WN-WuP	-	0.212	-	0.261
SGNS-DEPS (concat+r)	-	0.539	-	0.399
Paragram+CF (cos)	-	0.346	-	0.453
Paragram+CF (mul+r)	-	0.386	-	0.439
SDSN	0.708	0.658	0.547	0.475
SDSN+SDF	0.722	0.671	0.562	0.495
SDSN+SDF+AS	**0.757**	**0.692**	**0.577**	**0.544**

Table 1: Graded lexical entailment detection results on the random and lexical splits of the HyperLex dataset. We report Spearman's ρ on both validation and test sets.

Since plenty of related research on lexical entailment is still focused on the simpler binary detection of asymmetric relations, we also run experiments on the large binary detection HypeNet dataset (Shwartz et al., 2016), where the SDSN output is converted to binary decisions. We again report scores for both random and lexical split.

Results and Analysis. The results on two HyperLex splits are presented in Table 1, along with the best configurations reported by Vulić et al. (2017). We refer the interested reader to the original HyperLex paper (Vulić et al., 2017) for a detailed description of the best performing baseline models.

The Supervised Directional Similarity Network (**SDSN**) achieves substantially better scores than all other tested systems, despite relying on a much simpler supervision signal. The previous top approaches, including the Paragram+CF embeddings, make use of numerous annotations provided by WordNet or similarly rich lexical resources, while for SDSN and **SDSN+SDF** only use the designated relation-specific training set and corpus statistics. By also including these extra training instances (**SDSN+SDF+AS**), we can gain additional perfor-

[2] Note that the lexical split discards all cross-set training-test word pairs. Consequently, the number of instances in each subset is lower than with the random split.

	Lexical split			Random split		
	P	R	F	P	R	F
Dual-T	70.5	78.5	74.3	93.3	82.6	87.6
HypeNet-hybrid	80.9	61.7	70.0	91.3	**89.0**	90.1
H-feature	70.0	**96.4**	81.1	92.6	85.0	88.6
SDSN	**82.8**	84.6	83.7	**94.0**	86.7	90.2
SDSN+SDF	82.6	86.0	**84.2**	92.8	88.7	**90.7**

Table 2: Results on the HypeNet binary hypernymy detection dataset.

mance and push the correlation to 0.692 on the random split and 0.544 on the lexical split of HyperLex, an improvement of approximately 25% to the standard supervised training regime.

In Table 3 we provide some example output from the final SDSN+SDF+AS model. It is able to successfully assign a high score to (*captain*, *officer*) and also identify with high confidence that *wing* is not a type of *airplane*, even though they are semantically related. As an example of incorrect output, the model fails to assign a high score to (*prince*, *royalty*), possibly due to the usage patterns of these words being different in context. In contrast, it assigns an unexpectedly high score to (*kid*, *parent*), likely due to the high distributional similarity of these words.

Glavaš and Ponzetto (2017) proposed a related dual tensor model for the binary detection of asymmetric relations (*Dual-T*). In order to compare our system to theirs, we train our model on HypeNet and convert the output to binary decisions. We also compare SDSN to the best reported models of Shwartz et al. (2016) and Roller and Erk (2016), which combine distributional and pattern-based information for hypernymy detection (*HypeNet-hybrid* and *H-feature*, respectively).[3] We do not include additional WordNet and PPDB examples in these experiments, as the HypeNet data already subsumes most of them. As can be seen in Table 2, our SDSN+SDF model achieves the best results also on the HypeNet dataset, outperforming previous models on both data splits.

5 Conclusion

We introduce a novel neural architecture for mapping and specialising a vector space based on limited supervision. While prior work has focused only on optimising individual word embeddings available in external resources, our model uses general-purpose embeddings and optimises a separate neural component to adapt these to the specific task, generalising to unseen data. The system achieves new state-of-the-art results on the task of scoring graded lexical entailment. Future work could apply the model to other lexical relations or extend it to cover multiple relations simultaneously.

[3]For more detail on the baseline models, we refer the reader to the original papers.

		S	P
captain	officer	8.22	8.17
celery	food	9.3	9.43
horn	bull	1.12	0.94
wing	airplane	1.03	0.84
prince	royalty	9.85	4.71
autumn	season	9.77	3.69
kid	parent	0.52	8.00
discipline	punishment	7.7	3.32

Table 3: Example word pairs from the HyperLex development set. S is the human-annotated score in the HyperLex dataset. P is the predicted score using the SDSN+SDF+AS model.

Acknowledgments

Daniela Gerz and Ivan Vulić are supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909). We would like to thank the NVIDIA Corporation for the donation of the Titan GPU that was used for this research.

References

Øistein Andersen, Julien Nioche, Edward J. Briscoe, and John Carroll. 2008. The BNC parsed with RASP4UIMA. In *LREC*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the ACL*, 5:135–146.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah a. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *NAACL-HLT*, pages 1606–1615.

Goran Glavaš and Simone Paolo Ponzetto. 2017. Dual tensor model for detecting asymmetric lexico-semantic relations. In *EMNLP*, pages 1757–1767.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Hans Kamp and Barbara Partee. 1995. Prototype theory and compositionality. *Cognition*, 57(2):129–191.

Douwe Kiela, Laura Rimell, Ivan Vulić, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *ACL*, pages 119–124.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Geoffrey Neil Leech. 1992. 100 million words of English: the British National Corpus (BNC).

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *ACL*, pages 302–308.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *NAACL-HLT*, pages 970–976.

Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. pages 142–148.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the ACL*, 5:309–324.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *NIPS*, pages 6341–6350.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *ACL*, pages 425–430.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Marek Rei and Ted Briscoe. 2014. Looking for hyponyms in vector space. In *CoNLL*, pages 68–77.

Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. Grasping the finer point: A supervised similarity network for metaphor detection. In *EMNLP*, pages 1537–1546.

Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *EMNLP*, pages 2163–2172.

Eleanor H. Rosch. 1975. Cognitive representations of semantic categories. *Journal of Experimental Psychology*, 104(3):192–233.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *EACL*, pages 38–42.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *ACL*, pages 2389–2398.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *EACL*, pages 65–75.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4).

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *COLING*, pages 2249–2259.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the ACL*, 3:345–358.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.